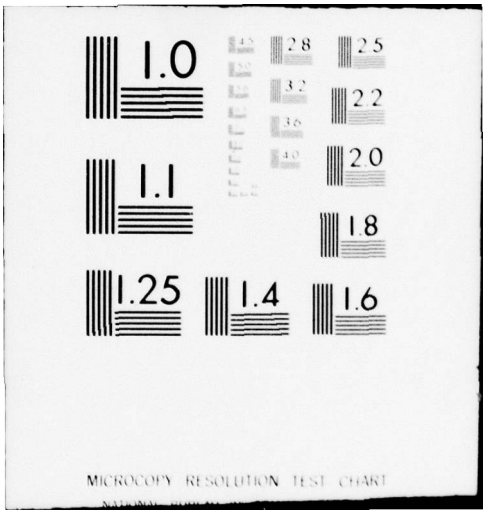
AD-A061 651 NORTH CAROLINA UNIV AT CHAPEL HILL F/G 9/2
SAMPLING FROM THE MULTINOMIAL DISTRIBUTION ON A COMPUTER.(U)
OCT 78 G S FISHMAN N00014-76-C-0302
UNCLASSIFIED TR-78-5 NL

1 OF 1
AD A061 651



END
DATE FILMED
2-79
DDC

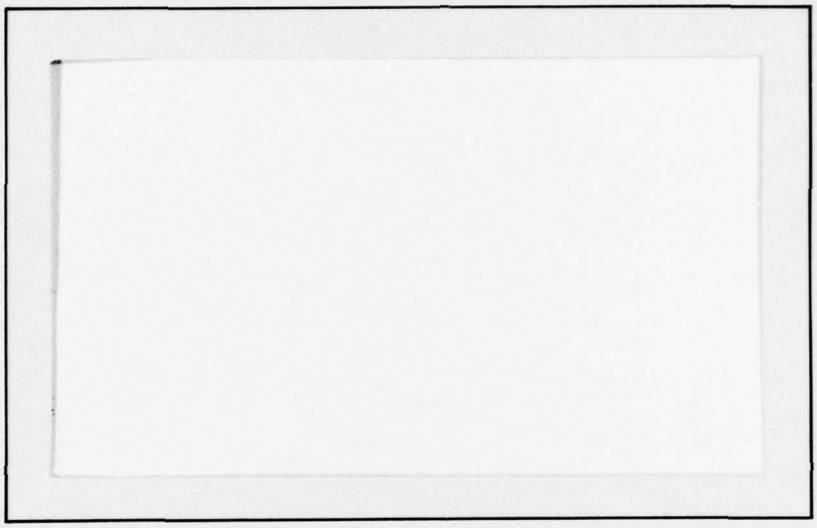

1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART

AD A061 651

DDC FILE COPY

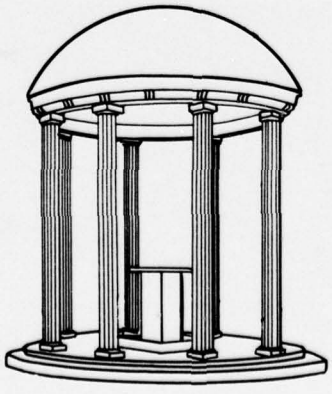
(12) LEVEL II

OPERATIONS RESEARCH AND SYSTEMS ANALYSIS



UNIVERSITY OF NORTH CAROLINA
AT CHAPEL HILL

DDC
RECEIVED
NOV 29 1978
A



DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

78 10 31 034

DDC FILE COPY AD A061 651

(6) SAMPLING FROM THE MULTINOMIAL DISTRIBUTION ON A COMPUTER.

(10) George S. Fishman

(9) Technical Report, No. (14) TR-78-5
(11) October, 1978

(12) 32p.

DDC
RECEIVED
NOV 29 1978
A

Curriculum in Operations Research
and Systems Analysis

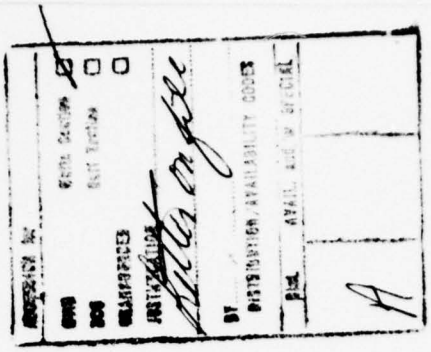
University of North Carolina at Chapel Hill

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

(15) This research was supported by the Office of Naval Research under contract N00014-76-C-0302.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

259 500 LB



Abstract

This paper describes algorithms to effect multinomial sampling on a computer in ways that protect the sampler against an excess computing cost per sample. Section 2 presents the multinomial model together with an equivalent representation in terms of a series of binomial sampling experiments. The binomial representation is further discussed in Section 4. Sections 3 and 5 demonstrate the danger of using too simplistic a sampling scheme if execution time is a concern. Section 6 describes how a normal approximation to the binomial distribution can make execution time virtually independent of n. A criterion of acceptability is described.

Section 7 describes an acceptance-rejection technique that, when used with a Poisson sample, allows the desired binomial sampling exactly. Using a normal approximation to the Poisson distribution, one can again generate multinomial samples virtually independent of n. A criterion of acceptability is described. The method of Section 7 may apply with given accuracy in cases in which the normal approximation to the binomial distribution does not apply.

Section 8 describes a procedure for binomial sampling based on the inverse transform method. Although the mean execution time is proportional to n, the procedure is intended for small n. Section 9 describes the ordering of the serial binomial experiments in response to alternative objectives. Section 10 describes algorithm M3 which puts all the suggestions of earlier sections together. The section also describes procedures for reducing the expense of testing to see if cost-saving generation methods apply.

78 10 31 034

1. Introduction

Multinomial sampling is an inherent feature of many computer based simulations. The most notable types of simulation that rely on this form of sampling have to do with population growth. For example, given n individuals in an initial state, one may wish to determine the numbers that pass to new states $1,\ldots,k$ in a unit time period. Here p_i, the probability of passing to state i, may be constant for the tenure of the simulation run or may change in each unit time period in response to environmental change. Let X_i denote the number that pass to state i. Then in a given year with specified n and $p_1,\ldots,p_k$, $X_1,\ldots,X_k$ form a multinomial sample. Although this form of sampling is hardly rare, little if any discussion has appeared in the published literature regarding how to perform this sampling efficiently on a computer. This paper addresses the problem of computational efficiency and describes algorithms to effect multinomial sampling in ways that protect the sampler against an excess cost per sample.

Section 2 presents the multinomial model together with an equivalent representation in terms of a series of binomial sampling experiments. The binomial representation is further discussed in Section 4. Sections 3 and 5 demonstrate the danger of using too simplistic a sampling scheme if execution time is a concern. Section 6 describes how a normal approximation to the binomial distribution can make execution time virtually independent of n. A criterion is presented there for determining when the approximation applies with a specified upper bound on absolute error.

The criterion is in terms of an inequality that easily can be built into a computer program.

Section 7 describes an acceptance-rejection technique that, when used with a Poisson sample, allows the desired binomial sampling exactly. Using a normal approximation to the Poisson distribution, one can again generate multinomial samples virtually independent of n. A criterion of acceptability is described. The method of Section 7 may apply with given accuracy in cases in which the direct approximation of the binomial distribution does not apply.

Section 8 describes a procedure for binomial sampling based on the inverse transform method. Although the mean execution time is proportional to n, the procedure is intended for small n for which evidence in [4] indicates its appeal. Section 9 describes the ordering of the serial binomial experiments in response to alternative objectives. For example, if one samples in order of decreasing p_i, the odds that the normal approximation in Section 6 applies are enhanced. Section 10 describes algorithm M3 which puts all the suggestions of earlier sections together. The section also describes procedures for reducing the expense of testing to see if cost-saving generation methods apply.

2. The Multinomial Model

Consider a series of n independent trials on each of which just one of k mutually independent events $E_1, \ldots, E_k$ occurs. The event E_i occurs with probability p_i. Let X_i denote the number of time that E_i occurs. Then the joint probability mass function (p.m.f.) of $\underline{X} = (X_1, \ldots, X_k)$ is

$$(1) \qquad \text{pr}\,(X_i = x_i\ ;\ i = 1,\ldots,\ k) = n! \prod_{i=1}^{k} (p_i/n_i!)$$

$$0 < p_i,\quad 0 \le x_i \quad i = 1,\ldots,\ k \quad \sum_{i=1}^{k} p_i = 1 \quad \sum_{i=1}^{k} x_i = n.$$

Here $\underline{X}$ has the multinomial distribution denoted by $M(n, p_1,\ldots, p_k)$. As an example, consider a population of immature female elephants of the same age in a park preserve in Africa. During a given year each elephant experiences one of $k = 4$ events:

E_1 = dies.

E_2 = survives and remains immature.

E_3 = survives and matures.

E_4 = survives, matures and conceives.

Maturity means capable of conceiving. Since these categories are exhaustive $p_1 + p_2 + p_3 + p_4 = 1$.

Let $\bar{E}_i$ denote the occurrence of an event other than E_i and let

$$M_i = \sum_{\substack{j=1 \\ j\neq i}}^{k} X_j \qquad i = 1,\ldots,\ k$$

denote the number of trials on which $\bar{E}_i$ occurs. Here $\bar{E}_i$ has probability $1-p_i$ and X_i and M_i have the joint p.m.f.

$$(2) \qquad \text{pr}(X_i = x,\ M_i = n-x) = \binom{n}{x} p_i^x (1-p_i)^{n-x} \quad x = 0,\ldots,\ n.$$

But (2) indicates that X_i has the binomial distribution $B(n, p_i)$.

Suppose that events $E_1,\ldots, E_j$ occur $x_1,\ldots,x_j$ times, respectively, for $j < k$ and define

$$(3) \qquad Z_j = \sum_{i=1}^{j} X_i \ .$$

Then one can show that

$$(4) \qquad \mathrm{pr}(X_{j+1} = x \mid Z_j = z) = \binom{n-z}{x} \left(\frac{p_{j+1}}{1-q_j}\right)^{x} \left(1 - \frac{p_{j+1}}{1-q_j}\right)^{n-x} \qquad x = 0, \ldots, n-z$$

$$z = \sum_{i=1}^{j} x_i \qquad\qquad q_j = q_{j-1} + p_j$$

$$q_0 = 0 \qquad q_k = 1 \qquad\qquad j = 1, \ldots, k-1,$$

so that X_{j+1} given $Z_j = z$ is from $\mathcal{B}\left(n-z, \frac{p_{j+1}}{1-q_j}\right)$. As we show shortly these results play central roles in multinomial sampling.

3. Simple Random Sampling

Let us now recast the problem in the more familiar terms encountered in discrete event simulation. Suppose that a given state has a population size n and that each member has a probability p_i of going to state i for $i = 1, \ldots, k$. The problem then becomes one of determining X_i, the number of individuals that move from the given state to state i. More generally one wants to determine $\underline{X} = (X_1, \ldots, X_k)$.

Many methods come to mind for sampling $\underline{X}$ from $M(n, p_1, \ldots, p_k)$ on a computer. The simplest relies on independent random sampling of the n events. Let U_i be a uniform deviate on $(0,1)$ whose distribution is denoted by $U(0,1)$. Then algorithm M1 affects the desired sampling. Here $I_{(q_{j-1}, q_j]}(U_i)$ denotes

the indicator function

$$(5) \qquad I_{(q_{j-1},\, q_j]}(U_i) = \begin{cases} 1 & q_{j-1} < U_i \le q_j \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm M1 $(n,\underline{p})$

1. $X_i \leftarrow 0 \qquad i = 1,\ldots, k.$
2. $i \leftarrow 1.$
3. Sample U_i from $U(0,1)$.
4. $m \leftarrow j \cdot I_{(q_{j-1},\, q_j]}(U_i)$.
5. $X_m \leftarrow X_m + 1.$
6. If $i = n$ deliver $\underline{X} = (X_1,\ldots, X_k)$.
7. $i \leftarrow i + 1.$
8. Go to 3.

Let $T(B)$ denote the mean execution time of algorithm B. Then the mean time to sample $\underline{X}$ from M1 has the form $T(M1) = a_1 + nf(k,\underline{p})a_2$ where a_1 and a_2 depend on computer and programming considerations and $f(k,\underline{p})$ depends on k and $\underline{p} = (p_1,\ldots, p_k)$ through the algorithm selected for executing step 4. Procedures T3 and T4 in Fishman [5, Sec. 9.18] describe tabling algorithms based on Marsaglia [9] that make $f(k, \underline{p})$ independent of k. These procedures are highly efficient timewise but require more space than search algorithms such as procedures T1 and T2 in [5, Sec. 9.18] do. Since all these algorithms for executing step 4 have their greatest appeal when $\underline{p}$ is fixed throughout a simulation, a question remains regarding their suitability for a simulation in which state transition probabilities change as time elapses. More importantly the linear growth of $T(M1)$ with n makes M1 prohibitively expensive for large n.

4. Sequential Binomial Sampling

Among the alternatives to simple random sampling, most rely on some form of binomial sampling. Algorithm M2 offers a prototype. Let T(M2, B) denote the mean execution time for M2 using

Algorithm M2 (n,p)

1. $Z \leftarrow 0$.
2. $X_i \leftarrow 0$ for $i = 1, \ldots, k$.
3. $i \leftarrow 1$.
4. Sample X_i from $\mathcal{B}(n-Z, p_i/(1-q_{i-1}))$.
5. If $X_i = n-Z$, deliver $X_1, \ldots, X_k$.
6. $Z \leftarrow Z + X_i$.
7. If $i = k-1$, $X_k \leftarrow n - Z$ and deliver $X_1, \ldots, X_k$.
8. $i \leftarrow i+1$.
9. Go to 4.

Algorithm B (to be described) for sampling from $\mathcal{B}(N, p)$. Then

$$(6) \qquad T(M2, B) = a_3 + a_4 k + a_5 \sum_{i=1}^{k-1} E[D(n-Z_{i-1}, p_i', B)].$$

$$Z_0 = 0 \qquad p_i' = p_i/(1-q_{i-1}) \quad i = 1, \ldots, k-1$$

where $E[D(N, p, B)]$ is the mean sampling time for algorithm B given N and p, Z_{i-1} is defined in (3), E is the expectation operator and a_3, a_4 and a_5 depend on computer and programming considerations.

5. Bernoulli Trials

Notice that (6) always has a term proportional to k. However, the form of dependence on n depend on the selected binomial sampling algorithm. Ahrens and Dieter [1,2] and Fishman [5, Sec. 9.14] describe several alternative algorithms. If one uses independent Bernoulli trials to sample X from $B(N, p)$ as in algorithm BE, one has

$$D(N, p, BE) = b_1 + b_2 N \min(p, 1-p) + b_3 N. \tag{7}$$

Algorithm BE (N, p)

1. $X \leftarrow 0$.
2. $\bar{p} \leftarrow p$.
3. If $p > 0.5$ $\bar{p} \leftarrow 1-p$.
4. $i \leftarrow 1$.
5. Sample U from $U(0,1)$.
6. If $U \leq \bar{p}$, $X \leftarrow X+1$.
7. If $i < n$, $i \leftarrow i+1$ and go to 6.
8. If $p \leq 0.5$ deliver X.
9. $X \leftarrow n-X$.
10. Deliver X.

The quantities b_1, b_2 and b_3 depend on computer and programming considerations. The quantity b_3 denotes the time spent executing a trial, regardless of its outcome, whereas b_2 denotes the time spent incrementing X when a success occurs. Generally $b_3 >> b_2$. Steps 2, 3, 8 and 9 also require explanation. One can easily show that if

X is from $\mathcal{B}(N, p)$ then N-X is from $\mathcal{B}(N, 1-p)$. By using $\bar{p} = \min(p, 1-p)$, the mean number of additions in step 5 is $N\bar{p} \leq Np$. For example, $p = 0.99$ gives $N\bar{p} = N \times 0.1$ versus $Np = N \times 0.99$. Although the cost of addition is small, nevertheless the saving is worth making.

If one follows M2 then the mean time to sample X_i is

$$(8) \quad E[D(n-Z_{i-1}, p_i', BE)] = b_1(1-q_{i-1}^n) + b_2 n \min(p_i, 1-q_i) + b_3 n(1-q_{i-1}).$$

The possibility of not executing step 4 because $Z_{i-1} = n$ contributes the coefficient $1-q_{i-1}^n$ to b_1. One now can show that

$$(9) \quad T(M2, BE) = a_3 - a_5 b_1 + (a_4 + b_1)k - a_5 b_1 \sum_{i=1}^{k-2} q_i^n + n a_5 [b_2 \sum_{i=1}^{k-1} \min(p_i, 1-q_i) + b_3(k-1)p_k + b_3 \sum_{i=1}^{k-1} i\, p_i].$$

Since the term in n dominates, the Bernoulli method also loses appeal as n increases.

6. Normal Approximation

Provided that n is sufficiently large for a given $\underline{p}$, one can turn the large n to a virtue in multinomial sampling. It is a well known result in statistics that if X is from $\mathcal{B}(N, p)$ then $Y = (X-Np)/\sqrt{Np(1-p)}$ has a distribution that converges to the normal distribution $N(0,1)$ as N increases. Let Y be from $N(0,1)$. Then one computes X as[†]

$$(10) \quad X = \max [0, \min(N, \langle Y\sqrt{Np(1-p)} + Np + 0.5\rangle)] .$$

Algorithm BN describes how to effect this sampling.

[†]The quantity $\langle\theta\rangle$ denotes the largest integer in θ.

Algorithm BN (N, p)

1. $a \leftarrow Np$.
2. $b \leftarrow \sqrt{a(1-p)}$.
3. $a \leftarrow a + 0.05$.
4. Sample Y from N(0,1).
5. $X \leftarrow \max[0, \min(N, < Yb + a >)]$.
6. Deliver X.

For given p the error in treating X as binomial decreases as N increases. Assessment of this error goes as far back as Uspensky [10]. Here we use a result in Makabe [7] that provides a tighter error bound than Uspensky does. Let

(11) $\quad x - 1 = y\sqrt{Npq} + Np - 0.5 \qquad q=1-p, \quad p<q.$

Then for $Npq \geq 25$

(12) $\quad \Delta_1 \geq \left| pr\,(X \leq x-1) - pr\,(Y \leq y) \right|$

where

(13) $$\Delta_1 = \frac{q-p}{6\sqrt{2\pi Npq}} + \frac{0.053 + 0.027(p-q) + 0.054(p-q)^2}{Npq} + e^{-1.5\sqrt{Npq}}.$$

Table 1 shows the minimal N required to satisfy (13) for $\Delta_1 = 0.01$ and 0.02. The results were obtained by application of the Newton-Raphson method. We discuss the use of (13) in practice in Section 10.

For an X generated by alogrithm BN, one can show that the mean execution time of BN is

(14) $\quad D(N,p,BN) = b_4.$

If BN is used in step 4 of M2, one has

(15) $\quad T(M2, BN) = a_3 + a_4 k + a_5 b_4 (k-1).$

TABLE 1

Minimal N for Satisfying Absolute Error Δ_1

for Normal Approximation in (13)

p	$\Delta_1 = 0.01$	$\Delta_1 = 0.02$
0.001	68271	23640
0.002	34056	11801
0.003	22651	7855
0.004	16948	5882
0.005	13527	4698
0.006	11246	3909
0.007	9616	3345
0.008	8395	2922
0.009	7444	2594
0.010	6684	2331
0.020	3263	1147
0.030	2123	753
0.040	1554	556
0.050	1212	438
0.060	985	360
0.070	823	304
0.080	701	262
0.090	607	230
0.100	532	204
0.150	309	127
0.200	201	100
0.250	140	100
0.300	103	100
0.350	100	100
0.400	100	100
0.450	100	100
0.500	100	100

Since a little work shows that

$$\lim_{n\to\infty} T(M2, BN) = (a_3 + a_4 k + a_5 b_4), \tag{16}$$

clearly one can effect multinomial sampling relatively independent of n, provided that n is sufficiently large. The key components of this result are twofold. Firstly, the convergence of the binomial to the normal distribution is a necessary ingredient. Secondly, the independence of the cost of normal sampling of the binomial parameters in step 4 of BN is critical.

If one uses BN exclusively in M2, a serious error can arise in practice. Since successive calls to BN use n, $n-z_1$, $n-z_2$,..., $n-z_{k-2}$ it is entirely conceivable that (n, p_1), $(n-z_1, p'_2, \ldots, (n-q_i, p'_{i+1})$ for $i < k-1$ satisfy (12) for a given Δ but $(n-z_{i+1}, p'_{i+2}), \ldots, (n-z_{k-2}, p'_{k-1})$ do not. When the constraint on $n-z_i$ is not met, one needs an alternative sampling procedure to determine $X_{i+1}, \ldots, X_k$. Therefore, one cannot rely on BN to solve all needs. We describe alternatives in Sections 7 and 8.

7. Poisson Method

The problem one now faces is to find an efficient way of sampling from $B(N, p)$ when N does not satisfy a specified Δ_1 in (12) for a given p. Ahrens and Dieter [1] and [2] and Fishman [5, Sec. 9.14] and [4] describe alternative methods of binomial sampling. Among these the Poisson method in [4] and [5] seems most appealing for large N when BN does not apply. Let X be a Poisson distributed random

variable with p.m.f.

$$(17) \qquad pr(X = x) = \begin{cases} e^{-\mu}\mu^{x}/x! & x = 0, 1, \ldots \\ 0 & \text{otherwise,} \end{cases}$$

denoted by $\mathcal{P}(\mu)$. Let U be from $\mathcal{U}(0, 1)$. If

$$S_1: \quad X \le N$$

$$S_2: \quad U \le \frac{\langle\lambda\rangle!}{(N-x)!}\,\lambda^{N-X-\langle\lambda\rangle} \qquad \lambda \equiv \mu(1-p)/p$$

then X is from $\mathcal{B}(N, p)$. Since

$$pr(S_2|S_1) = \langle\lambda\rangle!/N!e^{\mu}(1-p)^{N}\lambda^{\lambda-N}\,pr(S_1),$$

the mean number of X's from (17) need to find an X from $\mathcal{B}(N, p)$ is

$$(18) \qquad C_N(\mu, p) = 1/pr(S_2|S_1)pr(S_1) = N!\,e^{\mu}(1-p)^{N}\lambda^{\lambda-N}/\langle\lambda\rangle!.$$

This expression is minimized by

$$(19) \qquad \mu^{*} = \begin{cases} N - \langle N(1-p)\rangle & N(1-p) - \langle N(1-p) \le p \\ p(\langle N(1-p)\rangle + 1)/(1-p) & \text{otherwise.} \end{cases}$$

See [5 , Sec. 9.14] or [4].

Now the properties of (18) are of seminal importance. Firstly,

$$(20) \qquad \lim_{N\to\infty} C_N(\mu, p) = 1/(1-p)^{1/2}.$$

Secondly, Table 2 shows $C_N(\mu^*, p) \times (1 - p)^{1/2}$ for selected N and p. The results reveals a virtual insensitivity of $C_N(\mu^*, p)$ to N and that $C_N(\mu^*, p) \sim 1/(1-p)^{1/2}$. Thirdly, if one makes use of the observation that $N - X$ is from $\mathcal{B}(N, 1-p)$ when X is from $\mathcal{B}(N, p)$, then one can replace p by $\bar{p} = \min(p, 1-p)$ everywhere in this section and use X if $p \leq 0.5$ and $N - X$ if $p > 0.5$. This third property allows

$$C_N(\mu^*, \bar{p}) \sim 1/(1-\bar{p})^{1/2} < \sqrt{2} \sim 1.41, \tag{21}$$

where $\bar{p}$ replaces p in (18).

Now the remaining cost of the Poisson method resides in the algorithm selected for Poisson sampling. Ahrens and Dieter [1], and [2] and Fishman [4] and [5 , Sec. 9.13] describe algorithms whose mean execution times are proportional to $(\mu^*)^{1/2}$. Since $\mu^* = N - \langle N(1-\bar{p})\rangle \sim N\bar{p}$ for large N, one can at least reduce the cost of binomial sampling and consequently of multinomial sampling to a factor proportional to $N^{1/2}$. This clearly improves on the linear dependence on N in random and Bernoulli sampling.

In practice, one also can improve on these square root methods considerably by exploiting the limiting behavior of $P(\mu)$. If X is from $P(\mu)$, then $(X-\mu)/\sqrt{\mu}$ has a distribution that converges to $N(0,1)$ as μ increases. The error or approximation has been studied by a number of writers, including Cheng [3] and Makabe and Morimura [8]. Here we use the result of Makabe and Morimura, who give the tighter bound. Let x be an integer and let y satisfy

$$x = y\sqrt{\mu} + \mu + 0.5. \tag{22}$$

Let Y be from $N(0,1)$. Then for $\mu \geq 1$

TABLE 2

$C_N(\mu^*, p) \times \sqrt{1 - p}$

N \ p	0.01	0.10	0.20	0.30	0.40	0.50
5	0.995	0.976	0.996	0.976	0.989	0.986
10	0.995	0.999	0.998	0.996	0.994	0.992
15	0.996	0.991	0.999	0.991	0.996	0.995
20	0.996	1.000	0.999	0.998	0.997	0.996
25	0.996	0.995	0.999	0.995	0.998	0.997
30	0.997	1.000	0.999	0.999	0.998	0.997
35	0.997	0.996	0.999	0.996	0.998	0.998
40	0.997	1.000	0.999	0.999	0.999	0.998
45	0.997	0.997	1.000	0.997	0.999	0.998
50	0.998	1.000	1.000	0.999	0.999	0.998
55	0.998	0.998	1.000	0.998	0.999	0.999
60	0.998	1.000	1.000	0.999	0.999	0.999
65	0.998	0.998	1.000	0.998	0.999	0.999
70	0.999	1.000	1.000	0.999	0.999	0.999
75	0.999	0.998	1.000	0.998	0.999	0.999
80	0.999	1.000	1.000	1.000	0.999	0.999
85	0.999	0.998	1.000	0.998	0.999	0.999
90	1.000	1.000	1.000	1.000	0.999	0.999
95	1.000	0.999	1.000	0.999	0.999	0.999
100	1.000	1.000	1.000	1.000	0.999	0.999
200	1.000	1.000	1.000	1.000	1.000	1.000
300	1.000	1.000	1.000	1.000	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
600	1.000	1.000	1.000	1.000	1.000	1.000
700	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000
900	1.000	1.000	1.000	1.000	1.000	1.000
1000	1.000	1.000	1.000	1.000	1.000	1.000

(21) $$\Delta_2 \geq |pr(X \leq x-1) - pr(Y \leq y)|$$

where

(22) $$\Delta_2 = \frac{1}{6\sqrt{2\pi\mu}} + 0.0544/\mu + 0.0108/\mu^{3/2} + 0.2743/\mu^2 + 0.0065/\mu^{5/2} + (1 + 0.5/\mu^{1/2})\, e^{-2\sqrt{\mu}} .$$

We return to this issue of specifying an upper bound on error in Section 10, when bringing the alternative binomial algorithms together in an omnibus multinomial sampling algorithm.

Algorithm BPN describes how to effect binomial sampling via the Poisson method using the normal approximation. Here step 19 tests S_2 and needs explanation. By working with $V = -\ln U$, which is an exponential deviate, one transforms S_2 to

(23) $$V = -\ln U \geq (\langle\lambda\rangle - N + X) \ln \lambda + \ln [(N-X)!] - \ln (\langle\lambda\rangle!) ,$$

from which step 19 follows. Also, by using a table of $I + 1$ entrees for $\ln(i!)$ for $i \leq I$ one improves efficiency. The choice of I is the user's. Steps 10 and 16 account for situations in which the tables are inadequate by using Stirling's approximation for large factorials. Apart from the time spent executing steps 10 and 16 when their conditions are true, algorithm BPN has mean execution time

Algorithm BPN (N, p)

Table: $f_1 \equiv 0$, $f_{i+1} = \sum_{j=1}^{i} \ln j$ for $i = 1, \ldots, I$.

$c \equiv \sqrt{2\pi}$.

1. $\bar{p} \leftarrow p$.
2. If $p > 0.5$, $\bar{p} \leftarrow 1-p$.
3. $r \leftarrow N(1 - \bar{p})$.
4. $s \leftarrow \langle r \rangle$.
5. $\mu^* \leftarrow N-s$.
6. If $r-s > \bar{p}$, $\mu^* \leftarrow \bar{p}(s+1)/(1-\bar{p})$.
7. $\lambda \leftarrow (1/\bar{p} - 1)\mu^*$.
8. $\phi \leftarrow \ln \lambda$.
9. $m \leftarrow \langle \lambda \rangle$.
10. If $m > I-1$, $q \leftarrow c - m + (m + 0.5) \ln m$ and go to 12.
11. $g \leftarrow f_{m+1}$.
12. $a \leftarrow (\mu^*)^{1/2}$.
13. Sample Y from N(0,1).
14. $X \leftarrow \max(0, \langle Ya + \mu^* + 0.5 \rangle)$.
15. $W \leftarrow N - X$.
16. If $W > I - 1$, $h \leftarrow c - W + (W + 0.5) \cdot \ln W$ and go to 18.
17. $h \leftarrow f_{W+1}$.
18. Sample V from E(1).
19. If $V \geq (m - W)\, \phi - g + h$, go to 13.
20. If $p \leq 0.5$ deliver X.
21. Deliver W.

$$(24)\quad E[D(N, p, BPN)] = b_5 + b_6\, c_N(\mu^*, p) \sim b_5 + b_6/\sqrt{\max(p, 1-p)}.$$

Then M2 with $n, p_1, \ldots, p_k$ and using BPN has mean time

$$(25)\quad T(M2, BPN) \sim a_3 + a_4 k + a_5 \left\{ b_5 \sum_{i=1}^{k-1} (1 - q_{i-1}^n) + b_6 \sum_{i=1}^{k-1} [(1 - q_{i-1})/\max(p_i, 1 - q_i)]^{1/2} \right\}$$

$$\leq a_3 + a_4 k + a_5 (b_5 + \sqrt{2}\, b_6)(k-1) .$$

8. Inverse Transform Method

There remains the issue of how to sample from the binomial distribution when neither the normal approximation of Section 6 nor the Poisson method of Section 7 applies. Timing experiments in [4] revealed that the inverse transform sampling method deserved serious consideration when $N\bar{p} < 15$. In fact, it was the least time consuming for most (N, p) combinations examined when compared with three alternative algorithms. Although one may justifiably consider using the best algorithm for each (N, p) combination, choosing the inverse transform method for all $N\bar{p} < 15$ allows for a simplicity of use for what [4, Table 2] shows to be a relatively modest increase in computing time.

Let X be a discrete random variable with

$$pr(X = x) = s_x \qquad t_x = t_{x-1} + s_x$$

$$0 < s_x < 1 \qquad t_{-1} = 0 \qquad x = 0, 1, \ldots .$$

Let U be a random variable from $U(0,1)$. Since

$$pr(t_{x-1} \le U < t_x) = \int_{t_{x-1}}^{t_x} du \qquad x = 0,1\ldots,$$

one can determine X as

$$X = \min\ (x : U \le t_x;\ x = 0,1,\ldots)\ .$$

For X from $B(N, p)$ one has

$$s_0 = (1 - p)^N \qquad s_{x+1}/s_x = \frac{N - x}{x + 1} \cdot \frac{p}{1 - p}\ .$$

Algorithm BI [4] describes the steps need to sample X from $B(N, p)$ using the inverse transform method. Here mean execution time

Algorithm BI (N,p)

A, B, C, D are double precision.

1. $q \leftarrow p$.
2. If $p > 0.5$ $q \leftarrow 1 - p$.
3. $s \leftarrow 1 - q$.
4. $A \leftarrow 1$.
5. $B \leftarrow q/s$.
6. $C \leftarrow (N + 1)B$.
7. $D \leftarrow A$.
8. $X \leftarrow 0$.
9. Sample U from $U(0, 1)$.
10. $V \leftarrow U/s^N$.
11. If $V \le A$ go to 16.
12. $X \leftarrow X + 1$.

13. $D \leftarrow D(C/X - B)$.

14. $A \leftarrow A + D$.

15. If $X < N$ go to 11.

16. If $p > 0.5$ go to 18.

17. Deliver X.

18. Deliver $N - X$.

has the form

(26) $$D(N, p, BI) = b_7 + b_8 N \min(p, 1 - p).$$

Then M2 with $n, p_1, \ldots, p_k$ and using BI has mean execution time

$$T(M2, BI) = a_3 + a_4 k + a_5 \left[b_7 \sum_{i=1}^{k-1} (1-q_{i-1}^n) + b_8 n \sum_{i=1}^{k-1} \min(p_i, 1-q_i) \right].$$

9. Ordering the States for Execution

As the previous sections show, viewing multinomial sampling as a sequence of binomial sampling experiments enables one to take advantage of BN, when it applies, then BPN when it applies and finally BI when $N \min(p, 1 - p)$ is relatively small. In practice, one wants to induce additional efficiency by thoughtfully choosing the order in which states $i = 1, \ldots, k$ are sampled. Let

(27) $$\underline{J} = (j_1, \ldots, j_k) \qquad 0 < j_i \leq k \qquad i = 1, \ldots, k$$

denote a vector of integers such that $j_i \neq j_\ell$ for $i \neq \ell$ and $i, \ell = 1,\ldots, k$. Here each j_i can assume values 1 through k but no two j_i's can be equal. The objective now is to pick $\underline{J}$ in an optimal way with regard to a specified norm.

Let us first concentrate on (13). Clearly, the larger Npq is, the more likely one is to satisfy a specified Δ. Prior to generating the sample $X_{j_1}, X_{j_2}, \ldots, X_{j_k}$, X_{j_i} has variance $Np_{j_i}(1-p_{j_i})$. One immediately desirable objective is to choose $\underline{J}$ so that

$$Np_{j_i}(1-p_{j_i}) \geq Np_{j_{i+1}}(1-p_{j_{i+1}}) \qquad i = 1,\ldots,k-1. \tag{28}$$

This follows from the assignment

$$p_{j_i} \geq p_{j_{i+1}} \qquad i = 1,\ldots,k-1. \tag{29}$$

The choice in (29) is also compatible with maximizing the odds of satisfying (22) for a specified Δ.

If neither BN or BPN apply, then one needs to reconsider the arrangement $\underline{J}$. Clearly the rule

$$p_{j_k} \geq p_i \qquad i = 1,\ldots, k$$

$$p_{j_k} \geq p_{j_i} \geq p_{j_{i+1}} \qquad i = 1,\ldots, k-2 \tag{30}$$

improves on (29) by assigning the potentially most costly sampling to the kth sampling position. Using the identity $X_{j_k} = N - X_{j_1} - X_{j_2} - \ldots - X_{j_{k-1}}$ avoids this cost.

10. Putting It All Together

Since alternative forms of binomial sampling appear attractive for different (N, p) pairs, one would like a multinomial sampling procedure for each (N, p) combination encountered. Algorithm M3 does this. Here Δ' and Δ'' are error upper bounds on (13) and (22) respectively. The reader should note that BN, BNP and BI are called separately rather than as components of one omnibus binomial procedure. This form is intentional. It reduces the amount of testing for choosing among procedures.

Algorithm M3 (N, p, Δ', Δ'')

Given $c = 1/6\sqrt{2\pi}$.

1. Determine $j_1, \ldots, j_k$ such that $p_{j_i} \geq p_{j_{i+1}}$ for $i = 1, \ldots, k - 1$.
2. $X_i \leftarrow 0$ for $i = 1, \ldots, k$.
3. $i \leftarrow 1$.
4. $q \leftarrow 0$.
5. $p \leftarrow p_{j_i}$.
6. $N \leftarrow n$.
7. $A \leftarrow 8$.
8. $m \leftarrow \sqrt{Np(1 - p)}$.
9. If $m < 5$ go to 12.
10. If Δ_I satisfies condition Δ', sample X_{j_i} from BN(N, p) and go to 21.
11. $A \leftarrow 11$.
12. $m \leftarrow \sqrt{Np}$.

13. If $m < 5$ go to 15.

14. If Δ_2 satisfies Δ'', sample X_{j_i} from BNP(N, p) and go to 21.

15. $A \leftarrow 19$.

16. $r \leftarrow j_i$.

17. $j_\ell \leftarrow j_{\ell+1}$ for $\ell = 1, \ldots, k-1$.

18. $j_k \leftarrow r$.

19. $p \leftarrow p_{j_i} / (1 - q)$.

20. Sample X_{j_i} from BI(N, p).

21. $N \leftarrow N - X_{j_i}$.

22. If $N = 0$, deliver $\underline{X}$.

23. If $i = k - 1$, $X_{j_k} \leftarrow N$ and deliver $\underline{X}$.

24. $q \leftarrow q + p_{j_i}$.

25. $i \leftarrow i + 1$.

26. $p \leftarrow p_{j_i} / (1 - q)$.

27. Go to A.

The reader will note the less than complete specification of tests (13) and (22) in steps 10 and 14 respectively. This is deliberate. One can use these tests or slightly more conservative tests that are considerably more computationally efficient.

For the normality criterion in (13), let

$$m = \sqrt{Npq} \qquad c_1 = (q - p)/6\sqrt{2\pi}$$

(31)

$$c_2 = 0.053 + 0.027(q - p) + 0.054(q - p)^2 \qquad c_3 = 1.5 .$$

Consider the condition

$$(32)\quad S:\quad \Delta' \geq \Delta_1 = \frac{c_1}{m} + \frac{c_2}{m^2} + e^{-c_3 m}$$

where Δ_1 is simply (13) rewritten in terms of (31). Transposing terms leads to

$$(33)\qquad \omega = \left(\Delta' - \frac{c_2}{m} - e^{-c_3 m}\right) \geq \frac{c_1}{m} .$$

Since $m^2 \geq 25$, in order for (13) to apply, and $q - p \leq 1$, it follows that $\omega \geq 0$ if

$$(34)\qquad \Delta' \geq \frac{0.134}{25} + e^{-7.5} = 0.005913 .$$

Assuming that the user specified Δ' satisfies (34), one can write (32) in the equivalent form

$$(35)\quad S:\quad \left(\Delta' - \frac{c_2}{m^2} - e^{-c_3 m}\right)^2 \geq \left(\frac{c_1}{m}\right)^2 .$$

Since $m \geq 5$, one has

$$(36)\qquad e^{-c_3 m} \leq e^{-7.5} = 0.000553$$

which leads to the slightly more conservative condition

$$(37)\quad S':\quad \left(\Delta' - \frac{c_2}{Npq} - 0.000553\right)^2 \geq \frac{{c_1}^2}{Npq} .$$

In short, using (37) in place of (32) in step 10 of algorithm M3 and restricting $\Delta' \geq 0.005913$ avoid the need to compute $m = \sqrt{Npq}$ and

$e^{-1.5m}$. In summary, one can avoid exponentiation and square root transformations as follows for $\Delta' \geq 0.005913$:

a. If $Npq < 25$, BN does not apply.

b. If $Npq \geq 25$ and (37) holds, use BN.

A more conservative, but more computationally efficient, test than the criterion (22) implies is also possible for approximating the Poisson distribution by the normal. Let

$$h = \sqrt{\mu} \qquad \gamma(h) = \left(1 + \frac{1}{2h}\right)e^{-2h}$$

$$g_1 = \frac{1}{6\sqrt{2\pi}} \qquad g_2 = 0.0544 \qquad g_3 = 0.0108$$

$$g_4 = 0.2743 \qquad g_5 = 0.0065 .$$

For specified Δ'' one can write the criterion of interest as

$$\Delta'' \geq \Delta_2 = \frac{g_1}{h} + \frac{g_2}{h^2} + \frac{g_3}{h^3} + \frac{g_4}{h^4} + \frac{g_5}{h^5} + \gamma(h) . \tag{39}$$

If one adopts the lower bound $\Delta' \geq 0.005913$ in (34) for Δ'' as well, then $h \geq 12.1$, or equivalently $\mu \geq (12.1)^2 = 146.71$, guarantees (39).

Consider

$$h^* = \min\left[h: \quad \Delta'' \geq \frac{g_2}{h^2} + \frac{g_4}{h^4} + \gamma(h)\right] . \tag{40}$$

For $\Delta'' \geq 0.05913$ $h^* \sim 3.8$. When $h \geq h^*$ one can write (39) equiva-

lently as

$$(41) \qquad \left[\frac{\Delta'' h^4 - g_2 h^2 - g_4 - \gamma(h) h^4}{g_1 h^4 + g_3 h^2 + g_5} \right]^2 \geq \frac{1}{h^2} \quad .$$

Since one can show that $\gamma(h)h^4$ monotonically decreases for $h \geq h^*$, a slightly more conservative criterion than (41) is

$$(42) \qquad \mu \geq \left(\frac{g_1 \mu^2 + g_3 \mu + g_5}{\Delta'' \mu^2 - g_2 \mu - g_4 - 0.1181} \right)^2$$

where $\gamma(h)h^4 = 0.1181$ for $h = 3.8$.

For $h < h^*$ one has (39) equivalently as

$$(43) \qquad \left[\frac{\Delta'' h^4 - g_2 h^2 - g_4 - \gamma(h) h^4}{g_1 h^4 + g_3 h^2 + g_5} \right]^2 \leq \frac{1}{h^2}$$

since

$$(44) \qquad \Delta'' < \frac{g_2}{h^2} + \frac{g_4}{h^4} + \gamma(h) \quad .$$

Then a more conservative criterion than (43) is

$$(45) \qquad \mu \leq \left(\frac{g_1 \mu^2 + g_3 \mu + g_5}{\Delta'' \mu^2 - g_2 \mu - g_4} \right)^2 \quad .$$

Notice that (42) and (45) do away with the need to compute square roots and to exponentiate.

In summary, one can avoid exponentiation and square root transformations as follows for $\Delta'' \geq 0.005913$:

a. If $\mu \geq 146.71$ use BPN .

b. If $(3.8)^2 = 14.44 \leq \mu < 146.71$ and (42) holds, use BPN.

c. if $\mu < 14.44$ and (45) holds, use BPN.

REFERENCES

1. Ahrens, J. H. and U. Dieter, "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions," Computing, Vol. 12, 1974, pp. 223-246.

2. Ahrens, J. H. and U. Dieter, Non-Uniform Random Numbers, Institut fur Math. Statistik, Technische Hochschule in Graz, Austria, 1974.

3. Cheng, T. T., "The Normal Approximation to the Poisson Distribution and a Proof of a Conjecture of Ramanujan," Bulletin of the American Mathematical Society, Vol. 55, 1949, pp. 396-401.

4. Fishman, G. S., "Sampling from the Binomial Distribution on a Computer," Journal of the American Statistical Association, to appear.

5. Fishman, G. S., Principles of Discrete Event Simulation, John Wiley and Sons, 1978.

6. Johnson, N. L. and S. Kotz, Discrete Distributions, Houghton Mifflin, 1969.

7. Makabe, H., "A Normal Approximation to Binomial Distribution," Reports of Statistical Application Research, Union of Japanese Scientists and Engineers, Vol. 4, No. 2, August 1955, pp. 11-20.

8. Makabe, H. and H. Morimura, "On the Approximation to Some Limiting Distributions," Kōdai Mathematical Seminar Reports, Vol. 8, 1956, pp. 31-40.

9. Marsaglia, G., "Generating Discrete Random Variables in a Computer," Comm. ACM, Vol. 6, No. 1, 1968, pp. 25-28.

10. Uspensky, J. V., Introduction to Mathematical Probability, 1937, McGraw-Hill.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 78-5	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* SAMPLING FROM THE MULTINOMIAL DISTRIBUTION ON A COMPUTER		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) George S. Fishman		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0302
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of North Carolina Chapel Hill, N.C. 27514		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE October 1978 13. NUMBER OF PAGES 27
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Acceptance-Rejection Method
Bernoulli Sampling
Binomial Distribution
Multinomial Distribution
Normal Approximation
Poisson Distribution
Sampling

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This paper describes algorithms to effect multinomial sampling on a computer in ways that protect the sampler against an excess computing cost per sample. Section 2 presents the multinomial model together with an equivalent representation in terms of a series of binomial sampling experiments. The binomial representation is further discussed in Section 4. Sections 3 and 5 demonstrate the danger of using too simplistic a sampling scheme if execution time is a concern. Section 6 describes how a

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

normal approximation to the binomial distribution can make execution time virtually independent of n. A criterion of acceptability is described.

Section 7 describes an acceptance-rejection technique that, when used with a Poisson sample, allows the desired binomial sampling exactly. Using a normal approximation to the Poisson distribution, one can again generate multinomial samples virtually independent of n. A criterion of acceptability is described. The method of Section 7 may apply with given accuracy in cases in which the normal approximation to the binomial distribution does not apply.

Section 8 describes a procedure for binomial sampling based on the inverse transform method. Although the mean execution time is proportional to n, the procedure is intended for small n. Section 9 describes the ordering of the serial binomial experiments in response to alternative objectives. Section 10 describes algorithm M3 which puts all the suggestions of earlier sections together. The section also describes procedures for reducing the expense of testing to see if cost-saving generation methods apply.